IN THE SPECIFICATION

Amendments to the Specification:

Please amend the paragraph beginning on page 29, line 7, as follows:

After completing the state upgrade, the J2EE server 204 takes the responsibility of upgrading application part of the EJB module 302a, in operation 915. This includes the operations of loading new logical schema for upgraded EJB module 302b ~~304b~~, in operation 916, loading new physical schema for upgraded EJB module 302b, in operation 917, disabling existing EJB module 302a, in operation 918, and enabling the upgraded EJB module 302b ~~304b~~, in operation 919. Advantageously, the embodiments of the present invention allow online upgrades to the managed state of a Java application. Further, the embodiments of the present invention allow upgrades to the application without disrupting the functionality of the application during the online upgrading process.

Please amend the paragraph beginning on page 23, line 20, as follows:

Figure 6 is a class diagram showing repository interfaces 600 used by the RSM, in accordance with an embodiment of the present invention. The repository maintains classes and configuration descriptors that represent partitioning of state for a J2EE application. The repository interfaces 600 includes EJB modules 302, a logical schema archives 602, a physical scheme archives 604, entity bean classes 606, dependent object descriptor 608, SMU archives 610, and state object classes 612.

Please amend the paragraph beginning on page 26, line 9, as follows:

In operation 712, the executive 202 requests the new control module 750b to continue the upgrade. In response, the new control module 750b upgrades all child J2EE

Servers, in operation 713. The new control module 750b also upgrades all application service modules, in operation 714, and upgrades all child applications by recursively using the application upgrade sequence 700, in operation 715 [[714]].

Please amend the paragraph beginning on page 1, line 11, as follows:

This application is related to (1) U.S. Patent Application No. 09/812,536 ~~(Attorney Docket No. SUNMP002A)~~, filed March 19, 2001, and "Method and Apparatus for Providing Application Specific Strategies to a Java Platform including Start and Stop Policies," (2) U.S. Patent Application No. 09/812,537 ~~(Attorney Docket No. SUNMP002B)~~, filed March 19, 2001, and entitled "Method and Apparatus for Providing Application Specific Strategies to a Java Platform including Load Balancing Policies," (3) U.S. Patent Application No. 09/833,845 ~~(Attorney Docket No. SUNMP003)~~, filed April 11, 2001, and entitled "Method and Apparatus for Performing Online Application Upgrades in A Java Platform," (4) U.S. Patent Application No. 09/833,856 ~~(Attorney Docket No. SUNMP004)~~, filed April 11, 2001, and entitled "Method and Apparatus for Performing Failure Recovery in a Java Platform," (5) U.S. Patent Application No.09/818,214 ~~(Attorney Docket No. SUNMP005)~~, filed March 26, 2001, and entitled "Method and Apparatus for Managing Replicated and Migration Capable Session State for A Java Platform," (6) U.S. Patent Application No. 09/825,249 ~~(Attorney Docket No. SUNMP006)~~, filed April 2, 2001, and entitled "Method and Apparatus for Partitioning of Managed State for a Java based Application," and (7) U.S. Patent Application No. 09/846,492 ~~(Attorney Docket No. SUNMP008)~~, filed April 30, 2001, and entitled "Method and Apparatus for Migration of Managed Application State for a Java based Application." Each of the above related application are incorporated herein be reference.

Please amend the paragraph beginning on page 18, line 15, as follows:

During application design, application designer partitions the replicated and migration capable state of an application using State Management Units of different types and State Partition. Such state partitioning may be specified using an application configuration descriptor or done dynamically through control module. The repository maintains a representation for the static partitioning of state using schema archives. Schema archives are described in greater detail in U.S. Patent Application No. 09/818,214 ~~(Attorney Docket No. SUNMP005)~~, filed March 26, 2001, and entitled "Method and Apparatus for Managing Replicated and Migration Capable Session State for A Java Platform," which is incorporated herein by ~~be~~ reference in its entirety.

Please amend the paragraph beginning on page 20, line 10, as follows:

The control module 550 is a part of a Java application that provides control and application-specific policies for the application. The control module 550 is described in greater detail in related U.S. Patent Application No. 09/812,536 ~~(Attorney Docket No. SUNMP002A)~~, filed March 19, 2001, and entitled "Method and Apparatus for Providing Application Specific Strategies to a Java Platform including Start and Stop Policies," which is incorporated by reference in its entirety. The control module 550 is responsible for supervising the J2EE server 204 and the EJB modules at application runtime. Since the RSM is part of application runtime for a J2EE server process 204, J2EE server 204 activates the RSM as part of the start J2EE server process 502, the upgrade module process 504, and the move module process 506. The control module 550 interacts with the J2EE server 204, which in turn drives uses cases for RSM.

Please amend the paragraph beginning on page 14, line 18, as follows:

As shown in Figure 2, the RSM subsystem 206 is part of the J2EE Application Runtime subsystem 202, which is responsible for management and supervision of running the J2EE application 208. The RSM 206 manages the replicated and migration capable state for J2EE applications 208 that are running on a J2EE server 204. By managing application state, the RSM 206 [[204]] provides support for online application upgrades, failure recovery and load balancing features of the J2EE system 200. The RSM 206 uses a memory database within a J2EE server process 204 to manage the application state. In this manner, the RSM enables an application to remain operational even if the state servers become temporary unavailable.

Please amend the paragraph beginning on page 19, line 1, as follows:

RSM partitions entity beans based on the range of primary keys that will be served by different state partitions. For ~~From~~ example, account 1 to 100 will be managed within a single state partition 1, while account 101 to 200 will be managed in a separate state partition 2. A state partition also identifies the unit of concurrency. J2EE server process and RSM serialize transactional access to a state partition. At any specific instance, there can be only one transaction active on entity beans within a state partition.

Please amend the paragraph beginning on page 19, line 15, as follows:

A SMU is a collection of state objects 314 with the same state management type 400, and further defines a unit of checkpoints for the recoverable state types 404. An application can include multiple SMUs of different types depending on the specific requirements of the application. The RSM replicates a disk replicated SMU 408 to a disk replicated state server 212. The RSM is then capable of automatically recovering the disk replicated SMU 408 during an application restart after failure, shutdown or migration.

Please amend the paragraph beginning on page 19, line 21, as follows:

The RSM replicates a memory replicated SMU 410 to a memory replicated state server 214, and is then capable of automatically recovering memory replicated SMU 410 during an application restart after failure, shutdown or migration. The not replicated SMU 406 is not replicated by the RSM to either a disk replicated state server or a memory replicated state server. Generally, the RSM manages a non-replicated SMU 406 to support the migration state of a J2EE application from one J2EE server process to another.

Please amend the paragraph beginning on page 26, line 1, as follows:

The J2EE Server 204 then creates a new instance 750b of the old control module 750a, in operation 706, and loads the new class files for the new control module 750b, in operation 707. After loading the new class files, the J2EE Server 204, in operation 708, disables requests to the old control module 750a, and enables requests to new control module 750b, in operation 709. In this manner, requests from other application modules or from external applications can now be directed to the new control module 750b. The J2EE Server 204 then deletes the old control module 750a, in operation 710, and reports completion to the executive 202, in operation 711.